(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 639 907 A1

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 93113119.7

(22) Date of filing: 17.08.93

(51) Int. Cl.6: H04L 9/32, H04L 9/08

(43) Date of publication of application:
22.02.95 Bulletin 95/08

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(71) Applicant: R3 Security Engineering AG
Zürichstrasse 151
CH-8607 Aathal (CH)

(72) Inventor: Nyberg, Kaisa, Dr.
Prinz Eugen-Strasse 18/6
A-1040 Wien (AT)
Inventor: Rueppel, Rainer A., Dr.
Bahnhofstrasse 242
CH-8623 Wetzikon (CH)

(74) Representative: Blum, Rudolf Emil Ernst et al
c/o E. Blum & Co
Patentanwälte
Vorderberg 11
CH-8044 Zürich (CH)

(54) Digital signature method and key agreement method.

(57) A digital signature method based on the discrete logarithm problem is provided that allows message recovery. The message x is transformed according to the rule e = x g-r mod p, where r is a secret value generated by the signer. A value y is then calculated according to the rule y = r + s e mod q, where s is the signer's secret key. The signature of x consists of the pair (e,y). The verifier recovers the message x according the the rule x = gy ke e mod p, where k is the signer's public key. The validation of x can be based on some redundancy contained in x. Alternatively, a conventional verification equation can be constructed by using the signature method together with a hash function H. In addition, a key agreement method based on the signature method is provided which establishes with a single transmission pass a shared secret key K between two parties A and B in an authenticated fashion.

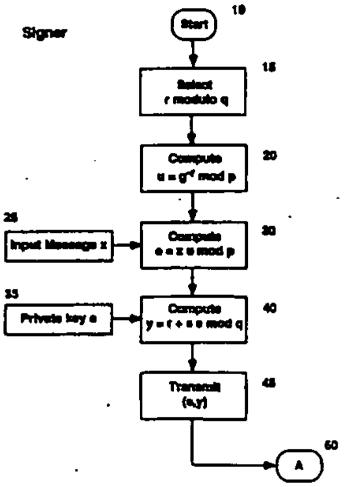

Figure 1

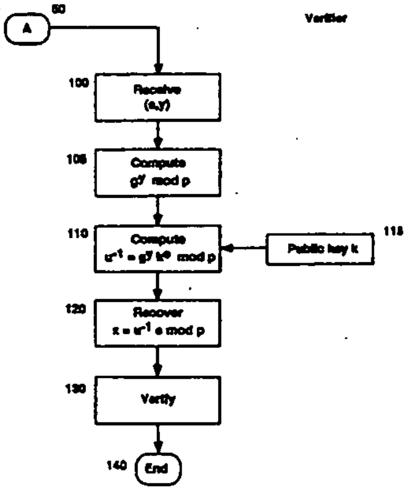

Figure 2

Rank Xerox (UK) Business Services
(3.10/3.09/3.3.4)

Background of the Invention

Field of the Invention

The invention relates to a method for generating and verifying a digital signature of a message. The field of this invention is data integrity and in particular generating and verifying a digital signature for a message or data file. The inventive method can also be used to establish a shared secret key between two parties.

The invention also relates to an apparatus for generating and/or verifying a digital signature.

Background Art

When a message is transmitted from one party to another, the receiving party may desire to determine whether the message has been altered in transit. Furthermore, the receiving party may wish to be certain of the origin of the message. It is known in the prior art to provide both of these functions using digital signature methods. Several known digital signature methods are available for verifying the integrity of a message. These known digital signature methods may also be used to prove to a third party that the message was signed by the actual originator. Several attempts have been made to find practical public key signature schemes that are based on the difficulty of solving certain mathematical problems to make alteration or forgery by unauthorized parties difficult. Most of the proposed schemes have been based either on the problem of factoring large integers or on the difficulty of computing discrete logarithms over finite fields (or over finite groups in general). For example, the Rivest-Shamir-Adleman system depends on the difficulty of factoring large integers (see "A method for obtaining digital signatures and public key cryptosystems", Communications of the ACM, Feb. 1978, Vol. 21, No. 2, pp. 120-126)

In 1985 Taher El-Gamal proposed a signature scheme based on the discrete logarithm problem (see "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Trans. on Inform. Theory, vol. IT-31, pp. 469-472, July 1985). In 1987 Chaum, Evertse and Van de Graaf proposed a zero-knowledge identification protocol based on the discrete logarithm problem. 1989 Schnorr proposed a modification of that protocol to obtain an efficient identification and signature scheme (see C.P. Schnorr, "Efficient Identification and Signatures for Smart Cards", Proceedings of Crypto'89, Springer-Verlag 1990, pp. 239-252 and European Patent Application EP 0 384 475 A1). 1991 the National Institute of Standards and Technology (NIST) proposed the "Digital Signature Method" that combines some features of ElGamal's and Schnorr's schemes (see Worldwide Patent WO 93/03562).

There are digital signature schemes that allow text recovery, that is, the original message can be recovered from the signature itself. Then it is unnecessary to send the message along with the signature. There are other signature schemes that do not allow text recovery but instead require the message or a hash value of the message in the verification process. All the described signature schemes based on the discrete logarithm problem have the property that from the signature the original message is no longer recoverable. With these systems it is necessary to send the message along with the signature. While any signature system that allows text recovery can be converted into a signature system with text hashing, the converse is not true.

Summary of the Invention

The goal of the present invention is therefore to provide an efficient digital signature method avoiding at least some of the disadvantages described above.

This digital signature method can be used in both message, recovery and message hashing mode. Clearly, when message recovery is possible, the original message need not be transmitted or stored together with the digital signature, which allows to improve the efficiency of the transmission or the storage.

This goal is reached by the method described in claim 1.

This method requires a pair of corresponding public and secret keys (k and s) for each signer.

In a preferred embodiment the message x is transformed according to the rule $e = x g^{-r} \bmod p$, where r is a secret value generated by the signer. A value y is then calculated according to the rule $y = r + s e \bmod q$. The signature of x consists of the pair (e,y) and is then transmitted. The receiving party of the signature uses a retransformation process to recover the message x. The received signature (e,y) is transformed according to the rule $x = g^y k^e e \bmod p$ thereby providing the original message x for legitimately executed signatures. The validation of x can be based on some redundancy contained in x. Alternatively, a conventional verification equation can be constructed by using a hash function H, transforming H(x) using the signature scheme and sending x along with the signature (e,y), recovering H(x) at the receiver using the retransformation equation and comparing it with the locally computed hash value of the received message x.

Such a signature system allows completely new applications, such as the key agreement method also described in the present invention.

This key agreement method establishes with a single transmission pass a shared secret key K between two parties A and B in an authenticated fashion. It requires that both parties have a pair of corresponding public and secret keys (k_A, s_A and k_B, s_B, respectively). Party A chooses a special key agreement message $x = g^R \bmod p$, signs it using the above signature method and transmits the resulting signature (e,y) to party B. Party B recovers the key agreement message using the retransformation process of the signature method. With a little additional computation both parties are now able to establish the shared key K. Party A computes $K = k_B{}^R \bmod p$ and party B computes $K = (g^R)^{-s_B} \bmod p$. Both arrive at the same value K since $k_B = g^{-s_B} \bmod p$.

Clearly, when text recovery is possible, the original message need not be transmitted or stored together with the digital signature, which allows to improve the efficiency of the transmission or the storage. Finally, a signature system with text recovery can always be run in text hashing mode without loss of efficiency, when the application demands signatures with text hashing.

Brief Description of the Drawings

Other advantages and applications of the inventive method will become apparent from the following description of preferred embodiments, wherein reference is made to the following figures:

Fig. 1 shows the signer's part of the digital signature method of the present invention,

Fig. 2 shows the verifier's part of the digital signature method of the present invention,

Fig. 3 shows the key agreement protocol based on the signature method of the present invention.

Detailed Description of the Invention

Referring now to Figs. 1, 2 where the digital signature method is shown.

Within the preferred digital signature method of the present invention each user has to obtain three numbers p, q and g. The value p is a large prime modulus with $p > 2^{512}$, the value q is a large prime divisor of (p-1) , and the value g is an element of multiplicative order q modulo p. That is, $g^t = 1 \bmod p$ if and only if t is an integer multiple of q. The triple (p,q,g) may be common to all users of the signature method, or may be chosen by each user independently.

The execution of the signature method begins at start terminal 10.

For every message a user wishes to sign, the user first selects secretly and randomly an integer r such that $0 < r < q$ (block 15).

Then in block 20 the value $u = g^{-r} \bmod p$ is calculated.

Block 25 denotes the input message x. In order to be recoverable by the receiver, the message x must be an integer between 0 and p. If the original message is larger than p, it can be subdivided into data blocks of size smaller than p. It is known in the art how to convert a message into a block representation where each block has a size smaller than some given integer value.

In block 30 the message x is transformed into value e using the equation

$$e = x\,u \bmod p$$

where u is the quantity already computed in block 20.Since the value u does not depend on the message it can be computed prior to knowledge of the message x. In general, e may be computed as $e = f(x,u)$ where the function f has the property that, given the values e and u, then the message x is easily recovered. One such function f is the multiplication of x times u modulo p shown in block 30. Other examples of such a function are $e = x + u \bmod p$ and $e = x \text{ xor } u$, where xor denotes the bitwise addition of x and u.

Block 35 denotes the private signature key s of the signer, where $0 < s < q$. The value of s is secretly chosen in advance to the execution of the signature method. It may be selected by the signer itself or by some trusted party which conveys s in a secret and authenticated way to the signer. The private signature key s is fixed for all messages to be signed by the signer.

The signature method proceeds to block 40, where the value of y is determined according to the rule

$$y = r + se \bmod q$$

The values e (determined in block 30) and y (determined in block 40) constitute the signature of message x. They are transmitted to the recipient as shown in block 45.

The connector 50 denotes the finishing of the signer's part and serves as reference for the continuation of the signature method at the recipient, i.e. the verifier.

After receiving the signature (e,y), as indicated in block 100, the recipient must recover the message x and verify the signature. For that purpose the recipient must know the values g, p and q used by the signer.

In block 105 the verifier computes the quantity $g^y \bmod p$.

Block 115 denotes the signer's public key k, which corresponds to the private signature key s through the rule $k = g^{-s} \bmod p$. This public key k and the identity of the signer must be made available in an authenticated fashion to the recipient of the signature (e,y). By possession of the public key k of the signer the verifier can then determine that the signature was originally created by that user who had knowledge of the private key s which corresponds to the particular value of k. If s has not been compromised, the signer's identity is linked to the public key y in an authenticated fashion.

In block 110 the verifier computes, using the signer's public key k, the inverse of the value u using the equation

$$u^{-1} = g^{y} k^{e} \bmod p$$

In block 120 the verifier recovers the message x using the equation

$$x = u^{-1} e \bmod p \qquad (1)$$

Block 120 shows the recovery operation when the function f to compute e from x and u was chosen to be multiplication modulo p (see block 30). The recovery transformation (1) will vary depending on the function f chosen for the calculation of e. For example, if f was chosen to be $e = x + u \bmod p$, then the recovery transformation (1) becomes $x = e - (u^{-1})^{-1} \bmod p$. Or, if f was chosen to be $e = x \text{ xor } u$ then the recovery transformation (1) becomes $x = e \text{ xor } (u^{-1})^{-1}$.

If the digital signature (e,y) was a genuine one and was received by the recipient in an unmodified or undistorted way, then equation (1) yields the correct value of the message x. If, however, the digital signature received was a forged one or was modified or distorted in any way during the transmission, then equation (1) will yield a different value x'. It will be understood by those skilled in the art that, by the nature of the retransformation equation (1), any redundancy contained originally in x will no longer be accessible in x'. Therefore, if immediate verification of the signature is desired, the verifier must inspect (block 130) the message x for the redundancy contained in it. The redundancy may be natural (for example, caused by the language in use) or artificial (for example, by some formatting rules imposed on x or by addition of some check values). If the redundancy check is successful, then the signature and the contained message x are accepted by the verifier as genuine. If the redundancy check fails, then the signature and the contained message x are rejected.

redundancy
[illegible]
[illegible]
artificial
such
check

It will be understood by those skilled in the art that the present invention can also be used in what is called the hashing mode. Then, instead of transforming the message x itself to yield the quantity e (as indicated in block 30), a hash value H(x) of the message x is used for the computation. A hash function H takes an arbitrary length message as input and yields a fixed length hash value as output. The hash value H(x) must now satisfy the requirement $0 < H(x) < p$, and x may have arbitrary length. The value of e is now determined as

$$e = H(x)\, u \bmod p$$

and in block 45 the message x has to be transmitted along with the signature, i.e. the triple (x, (e,y)) has to be transmitted to the recipient. The hash function H must be collision-resistant for the signature method to retain its qualities in hashing mode. That is, it must be computationally infeasible to find two messages x and x' such that $H(x) = H(x')\, g^{t} \bmod p$ for an arbitrary integer t.

The verifier proceeds as described until it recovers H(x) in block 120 (instead of x). To verify the signature, the verifier now applies the hash function H to the received message x and compares it with the recovered hash value from block 120. If both hash values are equal, the signature is accepted, if the two hash values differ, the signature is rejected.

It will be understood by those skilled in the art that the signs of certain values can be changed without changing the subject matter of this signature method. For example, the sign of the value r may be inverted such that in block 20 the value u is calculated according to the rule $u = g^{r} \bmod p$ and in block 40 the value y is calculated according to the rule $y = -r + se \bmod q$. Similarly, the sign of the value s may be inverted such that in block 40 the value y is calculated according to the rule $y = r - se \bmod q$ and in block 115 the public key k is congruent to $g^{s} \bmod p$.

Referring now to Fig. 3, there is shown the key agreement method of the present invention. For the key agreement method the two users A and B who wish to establish a shared secret key must use a common set of values p, q, and g as described in the signature method of the present invention.

The execution of the key agreement method begins at start terminal 210.

User A first selects secretly and randomly two integers R and r such that $0 < R, r < q$ (block 215).

Then in block 220 the value $e = g^{R-r} \bmod p$ is calculated.

Block 225 denotes the private key agreement key s_A of user A, where $0 < s_A < q$. The value of s_A is secretly chosen in advance to the execution of the key agreement method. It may be selected by user A itself or by some trusted party which conveys s_A in a secret and authenticated way to user A. The private key agreement key s_A is fixed for all executions of the key agreement method.

The key agreement method proceeds to block 230, where the value of y is determined according to the rule

$y = r + s_A e \bmod q.$

Block 235 denotes user B's public key agreement key k_B corresponding to the private key agreement key s_B through the rule $k_B = g^{-s_B} \bmod p$. User B's identity and public key k_B must be made available in an authenticated fashion to user A.

The key agreement method proceeds to block 240, where the value of the shared secret key K is determined according to the rule

$K = k_B^{R} \bmod p.$

The values e (determined in block 220) and y (determined in block 230) constitute the key agreement token. They are transmitted to the recipient as shown in block 245.

The connector 250 denotes the finishing of user A's part and serves as reference for the continuation of the key agreement method at the recipient, i.e. user B.

After receiving the key agreement token (e,y), as indicated in block 300, the recipient B must recover the value $g^R \bmod p$ and compute the shared secret key K.

In block 305 the verifier computes the quantity $g^y \bmod p$.

Block 315 denotes user A's public key agreement key k_A corresponding to the private key agreement key s_A through the rule $k_A = g^{-s_A} \bmod p$. User A's identity and public key k_A must be made available in an authenticated fashion to user B.

In block 310 the recipient B computes, using the sender's public key k_A, the quantity $k_A^{e} \bmod p$.

In block 320 the recipient B recovers the value $g^R \bmod p$ using the equation

$g^R = g^y k_A^{e}\, e \bmod p$

Block 325 denotes the private key agreement key s_B of user B, where $0 < s_B < q$. The value of s_B is secretly chosen in advance to the execution of the key agreement method. It may be selected by user B itself or by some trusted party which conveys s_B in a secret and authenticated way to user B. The private key agreement key s_B is fixed for all executions of the key agreement method.

The key agreement method proceeds to block 330, where the value of the shared secret key K is determined according to the rule

$K = (g^R)^{-s_B} \bmod p.$

Note that the same key K results from different computations at the sender A and at the recipient B, since

$$\begin{aligned} K &= k_B^{R} \bmod p \\ &= g^{-s_B\, R} \bmod p \\ &= (g^R)^{-s_B} \bmod p \end{aligned}$$

The key K is only known to the sender A and the receiver B since at both sides a secret value was used: sender A used the secret value R and recipient B used the secret value s_B. The key K is also authenticated to both sender and recipient, since the key token (e,y) transmitted from A to B was actually A's signature of the message $g^R \bmod p$ and since B used the private key agreement key s_B to compute the shared key K.

It will be understood by those skilled in the art that the signs of certain values can be changed without changing the subject matter of this key agreement method. For example, the sign of the value r may be inverted such that in block 220 the value e is calculated according to the rule $e = g^{R+r} \bmod p$ and in block 230 the value y is calculated according to the rule $y = -r + s_A e \bmod q$. Similarly, the sign of the value R may be inverted such that in block 220 the value e is calculated according to the rule $e = g^{-R-r} \bmod p$, in block 240 the value K is calculated according to the rule $K = k_B^{-R} \bmod p$, in block 320 the value $g^{-R} \bmod p$ is recovered, and in block 330 the value K is calculated according to the rule $K = (g^{-R})^{-s_B}$. Also similarly, the signs of the private keys s_A and s_B may be inverted such that in block 230 the value y is calculated according to the rule $y = r - s_A e \bmod q$, in block 330 the value K is calculated according to the rule $K = (g^R)^{-s_B}$, and in blocks 235 and 315, A's and B's public keys k_A and k_B are chosen to be congruent to $g^{s_A} \bmod p$ and $g^{s_B} \bmod p$, respectively. All these sign changes are independent of each other. They can be combined as desired.

The signature method and the key agreement method have been described in the setup of a finite field defined by arithmetic modulo p, also called the Galois Field with p elements and denoted GF(p). In the multiplicative group of GF(p) the discrete logarithm problem is difficult to solve. It will be understood by those skilled in the art that there are other cyclic groups which can equivalently be used as the basis for the present invention. For example, the extension field $GF(p^n)$ defined by arithmetic modulo an irreducible polynomial of degree n with coefficients modulo p or the cyclic group defined by an elliptic curve over a finite field could be used as the setup for the present invention. In principle, any cyclic group in which the discrete logarithm problem is difficult to solve, may serve as a basis for the present invention.

The inventive signature method and the key agreement method can be implemented in software and/or hardware by a person skilled in the art. An apparatus can e.g. be provided for creating the signature of a given message and transferring it to a second apparatus for verifying and retrieving the message.

Although the present invention has been shown and described with respect to specific preferred embodiments and variants thereof, it will be apparent that changes and modifications can be made without departing from what is regarded the subject matter of this invention.

Claims

1. A method for generating and verifying a digital signature e,y of a message x, comprising the following steps for generating the signature:

   a. providing a secret and random value r;
   b. providing a public value g;
   c. calculating a corresponding value u proceeding from a prime modulus p according to the rule

   $$u = g^{-r} \bmod p;$$

   d. calculating said value e from said message x and said value u according to the rule $e = f(G(x), u)$, wherein G(x) is a value derived from said message x and f is such that G(x) can be calculated from e and u using a function $h(u^{-1}, e) = G(x)$;
   e. calculating said value y proceeding from a value q, selected to be a divisor of p-1, according to the rule

   $$y = r + se \bmod q$$

   where s is a secret value;

   said method further comprising the following steps for verifying said signature e,y:
   f. calculating the inverse of said value u according to the rule

   $$u^{-1} = g^{y} k^{e} \bmod p$$

   where k is congruent to g^{-s} mod p and said value s is a secret value;
   g. reconstructing G(x) from u^{-1} and e according to the rule

   $$G(x) = h(u^{-1}, e)$$

   h. verifying the validity of said signature e,y.

2. The method of claim 1, wherein G(x) = x and wherein step h comprises the verification of a natural or artificially inserted redundancy in x.

3. The method of claim 1, wherein G(x) is a hash value H(x) computed by applying a hash function H to said message x and wherein step h comprises the verification of said signature by comparison of G(x) as obtained in step g with the value obtained from applying H to the message x directly.

4. The method of any of the preceding claims, wherein

   $$f(G(x), u) = G(x)\, u \bmod p \text{ and}$$
   $$h(u^{-1}, e) = u^{-1}\, e \bmod p,$$

   or wherein

   $$f(G(x), u) = G(x) + u \bmod p \text{ and}$$
   $$h(u^{-1}, e) = e - u \bmod p,$$

   or wherein

   $$f(G(x), u) = G(x) \text{ xor } u \text{ and}$$
   $$h(u^{-1}, e) = e \text{ xor } u.$$

5. The method of any of the preceding claims, wherein the sign of r and/or the sign of s are/is negative.

6. Method for generating and testing a digital signature e,y of a message x, especially according to any of the preceding claims, wherein the arithmetic modulo p is replaced by any other equivalent arithmetic, such as arithmetic in an extension field, arithmetic on an elliptic curve over a finite field, etc., where the discrete logarithm problem is difficult to solve.

7. A method for generating a digital signature according to any of the preceding claims.

8. Apparatus for carrying out the method of claim 7.

9. A method for verifying a digital signature according to any of the claims 1-6.

10. Apparatus for carrying out the method of claim 9.

11. A method for establishing a shared secret key K between two parties A and B, comprising the steps of:

a. providing to party A two secret and random values r and R;

b. providing a public value g;

c. calculating by party A the value e proceeding from a prime modulus p according to the rule

$$e = g^{R-r} \bmod p;$$

d. calculating the value y proceeding from a divisor q of p-1 according to the rule

$$y = r + s_A e \bmod q$$

where s_A is a secret value only known by party A;

e. calculating by party A said shared key K according to the rule

$$K = k_B{}^R \bmod p$$

where k_B is congruent to g^{-sB} mod p and said value s_B is a secret value only known by party B;

f. transmitting by party A to the recipient B the key token e,y containing said values e and y;

g. receiving by party B said key token e,y;

h. reconstructing by party B the value g^R mod p according to the rule

$$g^R = g^y k_A{}^e e \bmod p$$

where k_A is congruent to g^{-sA} mod p and said value s_A is a secret value only known by party A;

i. calculating by party B said shared key K according to the rule

$$K = (g^R)^{-sB} \bmod p$$

where s_B is a secret value only known by party B.

12. The method of claim 11, where in step c and d the sign of said value r is inverted, and/or where in step c, f, h and i the sign of said value R is inverted, and/or where in steps e, f, h and/or i the sign of said values s_A and s_B are inverted.

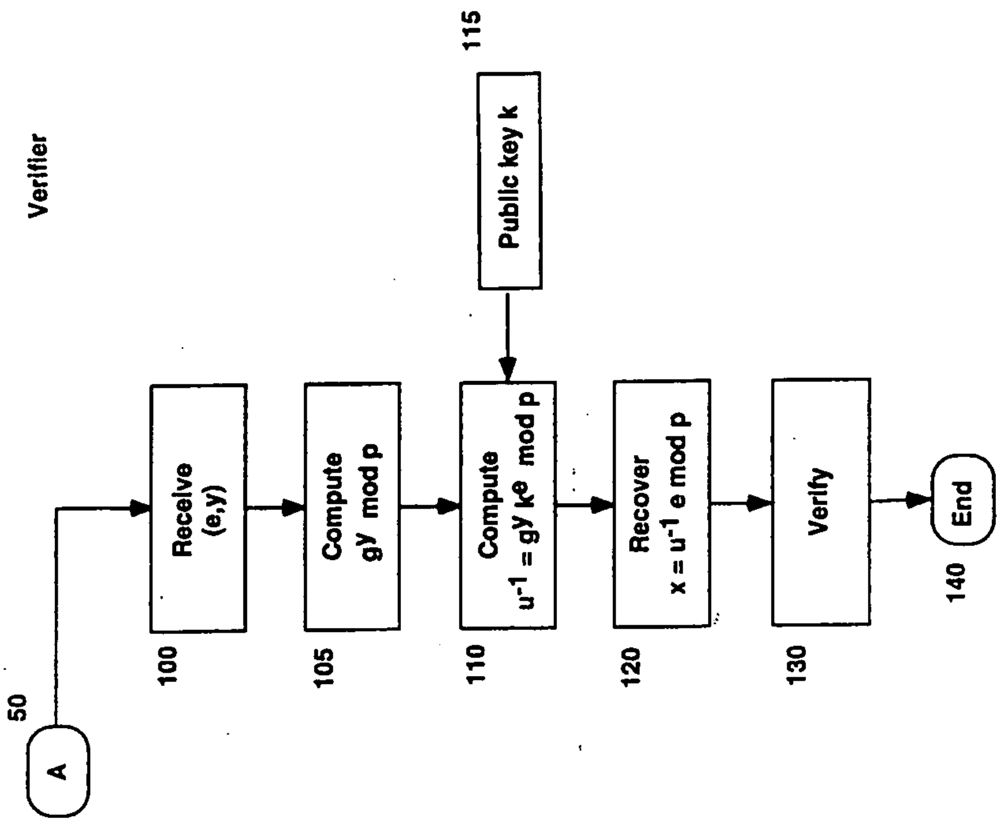

Figure 2

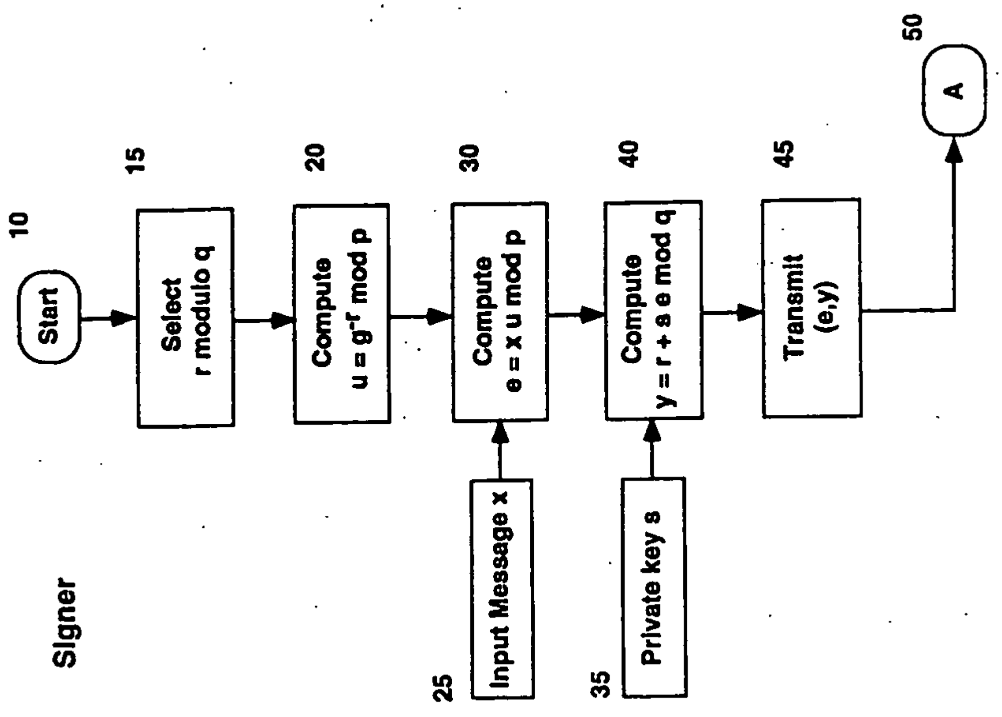

Figure 1

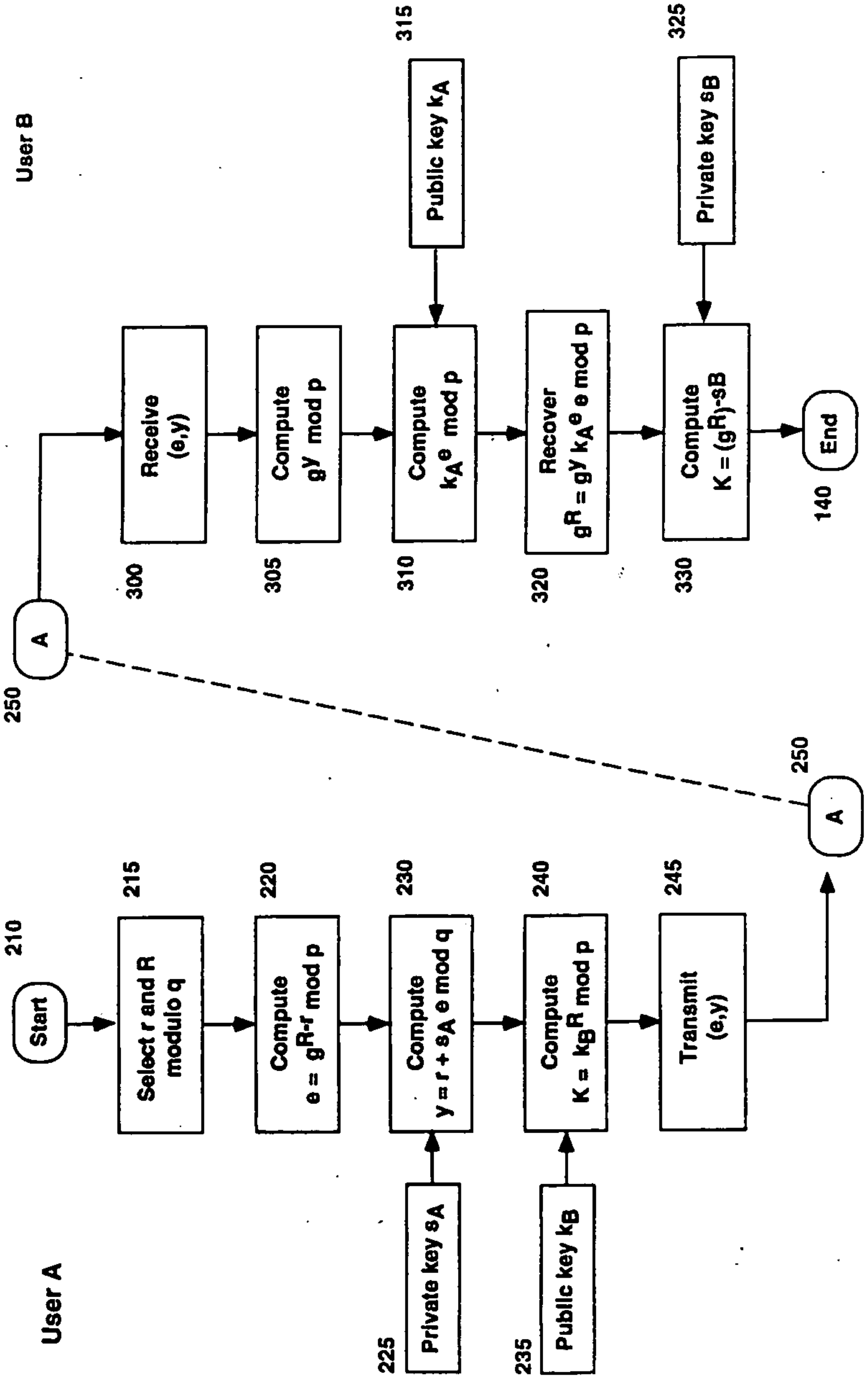
User A
User B
210
Start
215
Select r and R modulo q
220
Compute $e = g^{R-r} \bmod p$
225
Private key s_A
230
Compute $y = r + s_A e \bmod q$
235
Public key k_B
240
Compute $K = k_B^R \bmod p$
245
Transmit (e,y)
250
A
250
A
300
Receive (e,y)
305
Compute $g^y \bmod p$
310
Compute $k_A^e \bmod p$
315
Public key k_A
320
Recover $g^R = g^y k_A^e e \bmod p$
325
Private key s_B
330
Compute $K = (g^R)^{-s_B}$
140
End

Figure 3